

УДК 004.62

Студ. Грузинов О. А.

Науч. рук. асс. Савинко А. А.

(кафедра полиграфического оборудования и систем обработки информации, БГТУ)

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ УВЕДОМЛЕНИЯ ВЛАДЕЛЬЦЕВ ТРАНСПОРТНЫХ СРЕДСТВ

Введение. В современном мире достаточно много людей, обладающих автотранспортными средствами. Некоторые ставят на платную или бесплатную стоянку автомобиль и не испытывают проблем с парковкой. Но у большинства поблизости от места проживания нет парковочных мест и приходится время от времени сталкиваться с ситуациями, когда, например, на утро автомобиль заблокирован другим автомобилем и не предоставляется возможности выезда. Приходится различными способами искать владельца преграждающего путь автомобиля, на что уходит много времени, но и в этом случае результата можно так и не достичь, что грозит опозданием на работу, выговором и т.д. А ведь бывают и чрезвычайные ситуации, когда нужно экстренно освободить проезд. Для целей коммуникации владельцев автотранспортных средств создано *Android*-приложение.

Способ взаимодействия между отправителем и получателем сообщения. Для программного средства был выбран следующий способ взаимодействия *Android*-устройств через облачный сервис:

- *Android*-приложение подключается к облачному сервису *Firebase*;
- в *Android*-приложении создается сервис, который отвечает за принятие и обработку входящего сообщения от сервера *FCM (Firebase Cloud Messaging)*;
- на первом и втором устройстве происходит «подпись» на *topic*, при появлении новых данных в котором на устройство с сервера отправляется сообщение с этими данными, которые на устройстве отображаются в виде объекта;
- с *Android*-устройства отправителя сообщения отсылается *POST*-запрос на *Firebase Cloud Messaging* в формате *JSON*. Запрос распознается и заполняет *topic* данными пришедшими данными, после чего *FCM* отсылает сообщение на *Android*-устройство получателя сообщения;
- после того, как сообщение пришло на устройство получателя, на нём формируется и выводится уведомление для владельца. Вследствие такого способа взаимодействия и происходит обмен данными между устройствами при помощи облачного сервиса.

Технологии проектирования программного средства. При разработке данного приложения использовались различные технологии: формат передачи данных *JSON*, облачный сервис *Firebase*, *NoSQL* хранилище данных, язык программирования *Java*, платформа *Android*.

Основным компонентом программного средства является облачный сервис *Firebase*. *Firebase* предоставляет облачную *NoSQL* БД для *real-time* приложений в качестве сервиса. Данный сервис предоставляет *API* для разработчиков, позволяющий синхронизировать данные приложения между клиентами и хранить их в облаке *Firebase*.

За обработку сообщений отвечает сервис *FCM*. Это межплатформенная технология для обмена сообщениями, которая позволяет надежно доставлять сообщения. С помощью *FCM* можно уведомить клиентское приложение о том, что новая электронная почта или другие данные доступны для синхронизации. Для этого отправляются уведомления, чтобы управлять повторным подключением и удержанием пользователей. В случае использования, например, обмена мгновенными сообщениями, сообщение может передавать полезную нагрузку до 4 КБ в клиентское приложение. Работа технологии показана на схеме ниже на рисунке 1.

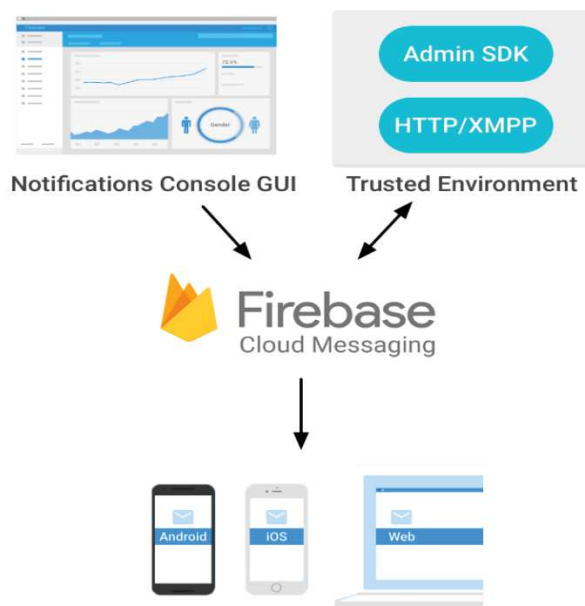


Рисунок 1 – Архитектура Firebase Cloud Messaging

Разработка программного средства. Для написания программного кода используется язык программирования *Java* на платформе *Android*. Ниже будет представлен фрагмент программного кода приложения, который продемонстрирует работу с уведомлениями.

```

public class FirebaseMessagingService extends
com.google.firebase.messaging.FirebaseMessagingService {

    public FirebaseMessagingService() {
    }
    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {
        // Check if message contains a data payload.
        if (remoteMessage.getData().size() > 0) {
            // Log.d(TAG, "Message data payload: " + remoteMessage.getData());
        }

        // Check if message contains a notification payload.
        if (remoteMessage.getNotification() != null) {
            String title = remoteMessage.getNotification().getTitle(); //get title
            String message = remoteMessage.getNotification().getBody(); //get message
            sendNotification(title, message);
        }
    }

    @Override
    public void onDeletedMessages() {
    }

    private void sendNotification(String title,String messageBody) {
        Intent intent = new Intent(this, AuthActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        PendingIntent pendingIntent = PendingIntent.getActivity(this, 0 /* Request code */, intent,
            PendingIntent.FLAG_ONE_SHOT);

        NotificationManager notificationManager =
            (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);

        Uri defaultSoundUri =
            RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
        NotificationCompat.Builder notificationBuilder = new NotificationCompat.Builder(this)
            .setSmallIcon(R.mipmap.ic_launcher)
            .setContentTitle(title)
            .setContentText(messageBody)
            .setAutoCancel(true)
            .setSound(defaultSoundUri)
            .setContentIntent(pendingIntent);
        notificationManager.notify(0 /* ID of notification */, notificationBuilder.build());
    }
}

```

Заключение. Программное средство позволит владельцам транспортных средств анонимно уведомлять друг друга в случае возникновения конфликтных ситуаций. Достаточно будет ввести государственный

номер автотранспортного средства и его владельцу немедленно будет выслано уведомление о том, что он кому-то заблокировал проезд.

УДК 655.3:004.434.5

Студ. Шаров Р. К.

Науч. рук. асс. Сулим П. Е.

(кафедра полиграфического оборудования и систем обработки информации, БГТУ)

ОБРАБОТКА ИЗОБРАЖЕНИЙ НА ЯЗЫКЕ PYTHON

Выбор в пользу того или иного языка программирования является следствием огромного количества факторов - от требований эффективно-го использования ресурсов вычислительной системы до наличия в нужное время подходящей книжки. Поэтому, чтобы избежать непродуктивного и спорного сравнения различных языков программирования друг с другом, рассмотрим только аргументы в пользу выбора языка Python.

1. Python - сравнительно молодой язык. Создавая его в 1990 - 1991 годах, его автор Гвидо ван Россум учёл все достоинства и недостатки предшествующих языков программирования.

2. Python имеет достаточно долгую историю развития и использования (почти 20 лет). В настоящее время Python поддерживается обширным международным сообществом разработчиков.

3. Python - развивающийся язык, используемый в реальных проектах.

4. Средства для работы с Python относятся к категории свободно распространяемого программного обеспечения (СПО). Это гарантирует во-первых, от каких-либо претензий относительно использования интеллектуальной собственности, а во-вторых, от превращения Python в обозримом будущем в мёртвый язык (например, как популярный Turbo Pascal).

Python прост в использовании. Базовая цель любого языка программирования – построить «мостик» между программистом и компьютером. Такие популярные языки как Visual Basic, C#, Java называют языками высокого уровня. Этот термин предполагает, что они ближе к человеческому языку, чем к машинному. Python с его простыми и ясными правилами еще приближается к английскому языку. Создание программ на Python настолько незамысловатый процесс, что о нем иногда говорят как о «программировании со скоростью мысли». С точки зрения профессионального программиста, легкость Python – залог высокой производительности труда. Программы на Python ко-